

In re Application of GOLDS et al.
Serial No. 09/768,098

REMARKS

The Office action has been carefully considered. The Office action rejected claims 11-13, 15-18, and 20 under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,870,756 to Nakata et al. ("Nakata"). Further, the Office action rejected claims 1-3, 5-10, 19 and 21-25 under 35 U.S.C. § 103(a) as being unpatentable over Nakata. Finally, the Office action rejected claims 4 and 14 under 35 U.S.C. § 103(a) as being unpatentable over Nakata in view of U.S. Patent No. 5,978,815 to Cabrera et al. ("Cabrera"). Applicants respectfully disagree.

By present amendment, claims 1, 11, 20, 21, and 24 have been amended for clarification and not in view of the prior art. Applicants submit that the claims as filed were patentable over the prior art of record, and that the amendments herein are for purposes of clarifying the claims and/or for expediting allowance of the claims and not for reasons related to patentability. Reconsideration is respectfully requested.

Applicants thank the Examiner for the interview held (by telephone) on January 19, 2005. During the interview, the Examiner and applicants' attorney discussed the claims with respect to the prior art. The essence of applicants' position is incorporated in the remarks below.

Prior to discussing reasons why applicants believe that the claims in this application are clearly allowable in view of the teachings of the cited and applied references, a brief description of the present invention is presented.

The present invention is directed to a system and method for ordering software modules in a persistent order for execution. To this end, the present

In re Application of GOLDS et al.
Serial No. 09/768,098

invention provides a mechanism whereby unique values may be statically assigned to software modules at the time that each of the software modules (e.g., filter drivers) may be developed. Each module's assigned value may determine its position relative to other modules in a stack. In this manner, the order for any given set of filter drivers may be fixed, eliminating bugs and other problems that result from alternative orderings, and also significantly simplifying testing.

In one implementation, this static value (sometimes referred to as an "altitude" because stacks are typically represented vertically) may comprise a precision floating-point number. As a result, when new software modules may be developed, each module may (in an existing execution order) be assigned a number that will enable that software module to be positioned between any two existing software modules, since between any two real numbers there exists an infinite number of other real numbers. By way of example, if altitudes such as 0.1 and 0.2 are assigned to filter drivers A and C, if some filter driver B is developed that needs to be ordered between A and C, there will an unused altitude available between A and C that can be assigned to B, e.g., 0.15. If some other filter needed to attach between B and C, there will always be an unused altitude between B and C (e.g., 0.18) that is available.

When applied to filter drivers, the drivers may be generally classified according to their type, e.g., (antivirus, quota, encryption), as it is already known where such classes should approximately attach. For example, if altitudes are assigned values in the range from 0.0 to 1.0, where higher values attach closer to the base file system (e.g., NTFS), antivirus products may be assigned an altitude in

In re Application of GOLDS et al.
Serial No. 09/768,098

the 0.2 to 0.3 range, quota drivers between 0.4 and 0.6, and encryption filters between 0.7 and 0.8. Moreover, drivers of the same type may also be ordered among one another within their general range, which may guarantee only one possible ordering in both testing and actual operation. Note that the above description is for example and informational purposes only, and should not be used to interpret the claims, which are discussed below.

§102 Rejections

Turning to the claims, amended claim 11 recites in a computer system, a mechanism comprising a plurality of software modules, each software module having a static assigned value indicative of a relative order, and an ordering mechanism configured to evaluate each static assigned value and to arrange the software modules for execution in a relative order determined by the assigned values, the order being deterministic and static.

The Office action rejected claim 11 as being anticipated by Nakata. More specifically, the Office action contends that Nakata teaches a plurality of software modules, each module having an assigned value indicative of a relative order and an ordering mechanism configured to evaluate each value and to arrange the software modules for execution in a relative order determined by the assigned values. Column 22, line 5 to column 23, line 57 of Nakata is referenced.

Applicants respectfully disagree.

Nakata is directed, generally, toward a system and method for maintaining and identifying file locations in an interchangeable storage media that may be docked into several different computer systems. Since different computer systems

In re Application of GOLDS et al.
Serial No. 09/768,098

utilize different file naming and file size conventions, the storage medium maintains information and data for a number of different naming and size conventions. In the embodiment cited in the Office action (column 22, lines 27-55), when a particular file is requested, a CPU of the computer system acquires a data file starting position that has been stored in correspondence with this file name and writes this data file in a RAM of the computer system (step 605). Next, the CPU reads the names of the group of stored data drivers used in association with the file name from the user data control information to the RAM (step 606). The CPU then reads the data driver control information into the RAM (step 607). Thus, any association with an assigned number is simply arbitrary and based solely on the nature of the data or the nature of the computer system into which the interchangeable storage media interfaces.

Contrarily, claim 11 recites each software module having a static assigned value indicative of a relative order. The static value, which may be assigned at the coding of the software module, cannot change without recoding the software module (which may not simply be done in practice since a new software module may also be coded to replace the old). Thus, the assigned values for indicating a relative order of execution may be static and persistent. This is different than Nakata in that assigned values (*i.e.*, start sectors as contended by the Office action) in Nakata will change with respect to which computer system is being interfaced.

Furthermore, claim 11 recites an ordering mechanism configured to evaluate each static assigned value and to arrange the software modules for execution in a

In re Application of GOLDS et al.
Serial No. 09/768,098

relative order determined by the assigned values, the order being deterministic and static. Once the software modules have been assembled, there may exist an order in which the software modules may be executed. This order may be deterministic (*i.e.*, known before-hand) and static (*i.e.*, will not change unless additional or different software modules are added). The system taught by Nakata, however, is completely silent as to ordering any execution of software modules. An example, given by Nakata describes executing software modules 1, 2, 3 and 4, but this is only coincidental in so much as the names of the sectors happen to be arbitrarily named 1, 2, 3, and 4 in this example. The names could have just as easily been New York, Seattle, Mickey Mouse, and James Dean as the order of software module execution is independent of the sector name. In fact, Nakata goes on to describe another example in which the sectors called for are 1, 5, 6, and 8. See column 23, lines 24-34 of Nakata. Thus, any order of execution taught by Nakata is simply not deterministic nor is it static. Nakata consequently fails to teach or suggest the recitations of claim 11. For at least these reasons, applicants submit that claim 11 is patentable over the prior art of record.

Applicants respectfully submit that dependent claims 12-13, 15-18, and 20, by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 11 and consequently includes the recitations of independent claim 11. As discussed above, Nakata fails to disclose or suggest the recitations of claim 11 and therefore these claims are also allowable over the prior art of record. In addition to the recitations of claim 11 noted above, each of these dependent claims includes additional patentable elements.

In re Application of GOLDS et al.
Serial No. 09/768,098

For example, claim 13 recites the mechanism of claim 11 wherein the software modules comprise filter drivers. The Office action contends that a filter driver reads on a data driver as taught by Nakata. This is flatly wrong. A filter driver may be a software module that contains executable language for allowing a computer system to work with a particular filter. As used in Nakata, a data driver is merely a software module for subjecting data files to processing so that the data files may be used by a particular computer system. A filter driver and a data driver are not the same. For at least this additional reason, applicants submit that claim 13 is allowable over the prior art of record.

§103 Rejections

Turning to the first independent claim rejected under §103, amended claim 1 recites in a computer system, a method, comprising maintaining static assigned values in association with software modules, each software module having a static assigned value, the assigned values having a relative order and there being an unassigned value between any two assigned values, and executing the software modules in an order determined by each of the assigned values, the order being deterministic and static.

The Office action rejected claim 1 as being unpatentable over Nakata. More specifically, the Office action contends that Nakata teaches maintaining assigned values in association with software modules, each software module having a assigned value. FIG. 30 of Nakata is referenced. Further, the Office action contends that Nakata teaches the assigned values having a relative order, and executing the software modules in an order determined by each of the assigned

In re Application of GOLDS et al.
Serial No. 09/768,098

values. Column 22, line 5 to column 23, line 57 of Nakata is referenced. The Office action acknowledges that Nakata fails to teach having unassigned values between any two assigned values of software modules. However, the Office action contends that Nakata teaches data files are stored and accessed based on start positions and if there are unused start positions, there necessarily exist unassigned values. The Office action concludes that it would be obvious to one skilled in the art at the time of the invention that since the drivers are also assigned start sectors that they also have unassigned values between any assigned values similar to the data files. Applicants respectfully disagree.

To establish *prima facie* obviousness of a claimed invention, all of the claim recitations must be taught or suggested by the prior art; (*In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974)), and "all words in a claim must be considered in judging the patentability of that claim against the prior art;" (*In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)). Further, if prior art, in any material respect teaches away from the claimed invention, the art cannot be used to support an obviousness rejection. *In re Geisler*, 116 F.3d 1465, 1471, 43 USPQ2d 1362, 1366 (Fed Cir. 1997). Moreover, if a modification would render a reference unsatisfactory for its intended purpose, the suggested modification / combination is impermissible. See MPEP § 2143.01

As discussed above, Nakata is directed, generally, toward a system and method for maintaining and identifying file locations in an interchangeable storage media that may be docked into several different computer systems. Since different computer systems utilize different file naming and file size conventions, the storage

In re Application of GOLDS et al.
Serial No. 09/768,098

medium maintains information and data for a number of different naming and size conventions. As noted previously, in the embodiment cited in the Office action (column 22, lines 27-55), when a particular file is requested, the CPU acquires a data file starting position that has been stored in correspondence with this file name and writes this data file in the RAM (step 605). Next, the CPU reads the names of the group of stored data drivers used in association with the file name from the user data control information to the RAM (step 606). The CPU then reads the data driver control information into the RAM (step 607). Thus, an association with an assigned number is simply arbitrary and based solely on the nature of the data or the nature of the computer system into which the interchangeable storage media interfaces.

Contrarily, claim 1 recites maintaining static assigned values in association with software modules. The static value, which may be assigned at the coding of the software module, cannot change without recoding the software module. Thus, the assigned values for indicating a relative order of execution are static and persistent. This is different than Nakata in that assigned values (*i.e.*, start sectors as contended by the Office action) in Nakata will change with respect to which computer system is being interfaced. Consequently, Nakata simply does not teach or suggest maintaining static assigned values in association with software modules as recited in claim 1. Moreover, the Office action concludes that it would be obvious to one skilled in the art at the time of the invention that since the drivers are also assigned start sectors that they also have unassigned values between any assigned values similar to the data files. Such broad, conclusory statements do

In re Application of GOLDS et al.
Serial No. 09/768,098

not come close to adequately addressing the issue of motivation to combine, are not evidence of obviousness, and therefore are improper as a matter of law. *In re Dembiczak*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999).

Furthermore, claim 1 recites executing the software modules in an order determined by each of the assigned values, the order being deterministic and static. Once the software modules have been assembled, there may exist an order in which the software modules may be executed. This order may be deterministic (*i.e.*, known before-hand) and static (*i.e.*, will not change unless additional or different software modules are added). The system taught by Nakata, however, is completely silent as to ordering any execution of software modules. An example, given by Nakata describes executing software modules 1, 2, 3 and 4, but this is only coincidental in so much as the names of the sectors happen to be arbitrarily named 1, 2, 3, and 4 in this example. The names could have just as easily been New York, Seattle, Mickey Mouse, and James Dean as the order of software module execution is independent of the sector name. In fact, Nakata goes on to describe another example in which the sectors called for are 1, 5, 6, and 8. See column 23, lines 24-34 of Nakata. Thus, any order of execution taught by Nakata is simply not deterministic nor is it static. Therefore, one cannot possibly construe the teachings of Nakata to teach or suggest the recitations of claim 1. For at least these reasons, applicants submit that claim 1 is patentable over the prior art of record.

Claims 2-3 and 5-10 were rejected as unpatentable over Nakata and claim 4 was rejected as unpatentable over Nakata in view of Cabrera. Applicants

In re Application of GOLDS et al.
Serial No. 09/768,098

respectfully submit that dependent claims 2-10, by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 1 and consequently includes the recitations of independent claim 1. As discussed above, Nakata, whether considered alone or in any permissible combination with the prior art of record fails to teach or suggest the recitations of claim 1 and therefore dependent claims 2-10 are also allowable over the prior art of record. In addition to the recitations of claim 1 noted above, each of these dependent claims includes additional patentable elements.

For example, claim 3 recites the software modules comprise filter drivers, and wherein calling the software modules includes passing file system requests thereto. The Office action contends that a filter driver reads on a data driver as taught by Nakata. This is flatly wrong. A filter driver may be a software module that contains executable language for allowing a computer system to work with a particular filter. As used in Nakata, a data driver is merely a software module for subjecting data files to processing so that the data files may be used by a particular computer system. A filter driver and a data driver are not the same and cannot be construed to teach or suggest the recitations of claim 3. For at least this additional reason, applicants submit that claim 3 is allowable over the prior art of record.

Claim 14 was rejected as unpatentable over Nakata in view of Cabrera and claim 19 was rejected as unpatentable over Nakata. Each of these claims depends either directly or indirectly from claim 11 and consequently includes the recitations of independent claim 11. Applicants respectfully submit that dependent claims 14 and 19, by similar analysis, are allowable. As discussed above, Nakata fails to

In re Application of GOLDS et al.
Serial No. 09/768,098

teach or suggest the recitations of claim 11. Neither Nakata nor any other prior art of record, whether considered alone or in any permissible combination, teach or suggest the recitations of claim 11 and therefore dependent claims 14 and 19 are also allowable. In addition to the recitations of claim 11 noted above, each of these dependent claims includes additional patentable elements.

Turning to the next independent claim, amended claim 21 recites a computer-readable medium having computer-executable instructions, comprising maintaining static assigned values in association with filter drivers, each filter driver having an assigned value, the assigned values having a relative order and there being an unassigned value between any two assigned values, and executing the filter drivers in an order determined by each of the assigned values, the order being deterministic and static.

The Office action rejected claim 21 as being unpatentable over Nakata. More specifically, the Office action contends that Nakata teaches maintaining assigned values in association with filter drivers, each filter driver having a assigned value. FIG. 30 of Nakata is referenced. Further, the Office action contends that Nakata teaches he assigned values having a relative order, and executing the filter drivers in an order determined by each of the assigned values. Column 22, line 5 to column 23, line 57 of Nakata is referenced. The Office action acknowledges that Nakata fails to teach having unassigned values between any two assigned values of software modules. However, the Office action contends that Nakata teaches data files are stored and accessed based on start positions and if there are unused start positions, there necessarily exist unassigned values.

In re Application of GOLDS et al.
Serial No. 09/768,098

The Office action concludes that this disclosure makes obvious the recitations of claim 21. Applicants respectfully disagree.

Unlike Nakata, claim 21 recites maintaining static assigned values in association with filter drivers. The static value, which may be assigned at the coding of the software module, cannot change without recoding the software module. Thus, the assigned values for indicating a relative order of execution are static and persistent. This is different than Nakata in that assigned values (*i.e.*, start sectors as contended by the Office action) in Nakata will change with respect to which computer system is being interfaced. Consequently, Nakata simply does not teach or suggest maintaining static assigned values in association with filter drivers as recited in claim 21. Moreover, the Office action concludes that it would be obvious to one skilled in the art at the time of the invention that since the drivers are also assigned start sectors that they also have unassigned values between any assigned values similar to the data files. Such broad, conclusory statements do not come close to adequately addressing the issue of motivation to combine, are not evidence of obviousness, and therefore are improper as a matter of law. *In re Dembiczak*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999).

Furthermore, claim 21 recites executing the filter drivers in an order determined by each of the assigned values, the order being deterministic and static. That is, once the filter drivers have been assembled, there exists one and only one order in which the filter drivers may be executed and this order is deterministic (*i.e.*, known before-hand) and static (*i.e.*, will not change unless additional or different filter drivers are added). The system taught by Nakata,

In re Application of GOLDS et al.
Serial No. 09/768,098

however, is completely silent as to ordering any execution of software modules, let alone filter drivers. Therefore, one cannot possibly construe the teachings of Nakata to teach or suggest the recitations of claim 21. For at least these reasons, applicants submit that claim 21 is patentable over the prior art of record.

Applicants respectfully submit that dependent claims 22 and 23, by similar analysis, are allowable. Each of these claims depends directly from claim 21 and consequently includes the recitations of independent claim 21. As discussed above, Nakata fails to teach or suggest the recitations of claim 21 and therefore claims 22 and 23 are also allowable. In addition to the recitations of claim 21 noted above, each of these dependent claims includes additional patentable elements.

Turning to the last independent claim, amended claim 24 recites a method, comprising classifying software modules into groups based on types thereof, assigning each software module a static value based on its group, each assigned value having a relative order that is deterministic and static and there being an unassigned value between any two assigned values, and maintaining an association between each software module and its assigned value.

The Office action rejected claim 24 as unpatentable over Nakata. More specifically, the Office action contends that Nakata teaches classifying software modules into groups based on types thereof, assigning each software module a value based on its group, each assigned value having a relative order, and maintaining an association between each software module and its assigned value. Column 10, line 5 to column 11, line 24, column 11, line 65 to column 12, line 27, FIG. 6 and FIG. 8 of Nakata are referenced. The Office action acknowledges that

In re Application of GOLDS et al.
Serial No. 09/768,098

Nakata fails to teach having unassigned values between any two assigned values of software modules. However, the Office action contends that Nakata teaches data files are stored and accessed based on start positions and if there are unused start positions, there necessarily exist unassigned values. The Office action concludes that this disclosure makes obvious the recitations of claim 24. Applicants respectfully disagree.

Claim 24 recites assigning each software module a static value. Once again, Nakata is simply silent to the idea and concept of having static values that indicate a deterministic and static order of execution. The static value, which may be assigned at the coding of the software module, cannot change without recoding the software module. Thus, the assigned values for indicating a relative order of execution are static and persistent. This is different than Nakata in that assigned values (*i.e.*, start sectors as contended by the Office action) in Nakata will change with respect to which computer system is being interfaced. Consequently, Nakata simply does not teach or suggest maintaining static assigned values in association with filter drivers as recited in claim 24. Moreover, the Office action concludes that it would be obvious to one skilled in the art at the time of the invention that since the drivers are also assigned start sectors that they also have unassigned values between any assigned values similar to the data files. Such broad, conclusory statements do not come close to adequately addressing the issue of motivation to combine, are not evidence of obviousness, and therefore are improper as a matter of law. *In re Dembiczak*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999).

In re Application of GOLDS et al.
Serial No. 09/768,098

Furthermore, claim 24 recites an order that is deterministic and static. That is, once the software modules have been assembled, there exists one and only one order in which the software modules may be executed and this order is deterministic (*i.e.*, known before-hand) and static (*i.e.*, will not change unless additional or different filter drivers are added). Again, one cannot possibly construe the teachings of Nakata to teach or suggest the recitations of claim 24. For at least these reasons, applicants submit that claim 24 is patentable over the prior art of record.

Applicants respectfully submit that dependent claim 25 by similar analysis, is allowable. This claim depends directly from claim 24 and consequently includes the recitations of independent claim 24. As discussed above, Nakata fails to teach or suggest the recitations of claim 24 and therefore claim 25 is also allowable over the prior art of record.

For at least these additional reasons, applicants submit that all the claims are patentable over the prior art of record. Reconsideration and withdrawal of the rejections in the Office action is respectfully requested and early allowance of this application is earnestly solicited.

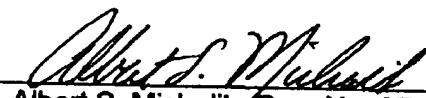
In re Application of GOLDS et al.
Serial No. 09/768,098

CONCLUSION

In view of the foregoing remarks, it is respectfully submitted that claims 1-25 are patentable over the prior art of record, and that the application is in good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,


Albert S. Michalik, Reg. No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE, Suite 193
Sammamish, WA 98074
(425) 836-3030
(425) 836-8957 (facsimile)

In re Application of GOLDS et al.
Serial No. 09/768,098

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this Amendment, along with transmittal, petition for extension of time, credit card payment form, and facsimile cover sheet, are being transmitted by facsimile to the United States Patent and Trademark Office in accordance with 37 C.F.R. 1.6(d) on the date shown below:

Date: April 19, 2005


Albert S. Michalik

2630 Amendment